Java I/O File Handling -

- 1. Write a program to create a new text file named test.txt.

```
package Assessement_day10;
import java.io.File;
import java.io.IOException;

public class create_file {

public static void main(String[] args) {
File file = new File("test1.txt");
try {
if (file.createNewFile()) {
System.out.println("File test.txt created successfully.");
} else {
System.out.println("File test.txt already exists.");
}
} catch (IOException e) {
System.out.println("Error creating file: " + e.getMessage());
}
}

}
```

Output:

File test.txt created successfully.

```
package Assesement_day9;
```

- 2. Write a program to check whether a file exists at a given path.

```
package Assessement_day10;
import java.io.File;
public class file_exist {

public static void main(String[] args) {
String filePath ="test.txt"; // replace with your file path
File file = new File(filePath);

if (file.exists()) {
System.out.println("File exists ");
} else {
System.out.println("File does not exist ");
}

}

}
```

Output:

File exists

3. Write a Java program to write "Hello, World!" into a

```
package Assessement_day10;
import java.io.FileWriter;
import java.io.IOException;
public class file_writer {
```

```
public static void main(String[] args) {
try (FileWriter writer = new FileWriter("hello.txt")) {
writer.write("Hello, World!");
System.out.println("Text written to file successfully.");
} catch (IOException e) {
System.out.println("Error writing to file: " + e.getMessage());
}
}

}
```

Output:

Text written to file successfully.

- 4. Write a program to read the content of a file line by line using BufferedReader.

```
package Assessement_day10;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class Buffer_reader {

public static void main(String[] args) {
StringBuilder content = new StringBuilder();
try (BufferedReader reader = new BufferedReader(new
FileReader("hello.txt"))) {
String line;
while ((line = reader.readLine()) != null) {
```

```
content.append(line).append("\n");
}
System.out.println(content.toString());
} catch (IOException e) {
System.out.println("Error reading file: " + e.getMessage());
}
}

}
```

Output:

Hello, World!

- 5. Write a program to append a line of text to an existing file.

```
package Assessement_day10;
import java.io.FileWriter;
import java.io.IOException;
import java.io.BufferedWriter;
public class append_line {
public static void main(String[] args) {
try (BufferedWriter writer = new BufferedWriter(new
FileWriter("hello.txt", true))) {
writer.write("This is a new line of text.");
writer.newLine();
System.out.println("Text appended to file successfully.");
} catch (IOException e) {
System.out.println("Error appending to file: " +
e.getMessage());
```

```
}
}

}
```

Output:

Hello, World!This is a new line of text.

- 6. Write a program to count the number of lines, words, and characters in a file.

```
package Assessement_day10;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class check_lines {

public static void main(String[] args) {
try (BufferedReader reader = new BufferedReader(new
FileReader("hello.txt"))) {
int lineCount = 0;
int wordCount = 0;
int charCount = 0;
String line;

while ((line = reader.readLine()) != null) {
lineCount++;
charCount += line.length() + 1; // +1 for newline character
wordCount += line.split("\\s+").length;
```

```
}

System.out.println("Lines: " + lineCount);
System.out.println("Words: " + wordCount);
System.out.println("Characters: " + charCount);
} catch (IOException e) {
System.out.println("Error reading file: " + e.getMessage());
}
}

}
```

Output:

```
Lines: 1
Words: 8
Characters: 41
```

- 7. Write a program to copy content from one file to another using FileReader and FileWriter.

```
package Assessement_day10;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Filewriter {

public static void main(String[] args) {
```

```
try (BufferedReader reader = new BufferedReader(new
FileReader("hello.txt"));
BufferedWriter writer = new BufferedWriter(new
FileWriter("destination.txt"))) {

String line;
while ((line = reader.readLine()) != null) {
writer.write(line);
writer.newLine();
}
System.out.println("File copied successfully.");

} catch (IOException e) {
System.out.println("Error copying file: " + e.getMessage());
}
}

}
```

Output:

File copied successfully.

8. Write a program to create, move, and delete a file using Files and Paths.

```
package Assessement_day10;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
```

```
public class move_file {

public static void main(String[] args) {
Path filePath = Paths.get("file1.txt");
Path newFilePath = Paths.get("newFile.txt");
Path movedFilePath = Paths.get("movedFile.txt");

try {
Files.createFile(filePath);
System.out.println("File created: " + filePath);

Files.write(filePath, "Hello, World!".getBytes());

Files.move(filePath, movedFilePath,
StandardCopyOption.REPLACE_EXISTING);
System.out.println("File moved to: " + movedFilePath);


Files.copy(movedFilePath, newFilePath,
StandardCopyOption.REPLACE_EXISTING);
System.out.println("File copied to: " + newFilePath);

Files.delete(newFilePath);
System.out.println("File deleted: " + newFilePath);
Files.delete(movedFilePath);
System.out.println("File deleted: " + movedFilePath);
} catch (IOException e) {
System.out.println("Error: " + e.getMessage());
}
```

```
}

}
```

Output:

```
File created: file1.txt
File moved to: movedFile.txt
File copied to: newFile.txt
File deleted: newFile.txt
File deleted: movedFile.txt
```

- 9. Write a program to read all lines of a file using Files.readAllLines() and print them.

```
package Assessement_day10;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
public class read_all_lines {

public static void main(String[] args) {
Path path = Paths.get("file.txt");

try {
Files.lines(path).forEach(System.out::println);
} catch (IOException e) {
System.out.println("Error reading file: " + e.getMessage());
}
```

```
}

}
```

Output:

```
Hello, World!
This is appended data.
```

- 10. Write a program to write data into a file using Files.write() and append using StandardOpenOption.APPEND.

```
package Assessement_day9;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;

public class Standard {

public static void main(String[] args) {
Path path = Paths.get("file.txt");
String data = "Hello, World!";

try {
Files.write(path, data.getBytes());
System.out.println("Data written to file successfully.");
String appendData = "\nThis is appended data.";
```

```
Files.write(path, appendData.getBytes(),
StandardOpenOption.APPEND);
System.out.println("Data appended to file successfully.");
} catch (IOException e) {
System.out.println("Error writing to file: " + e.getMessage());
}
}

}
```

Output:

```
Hello, World!
This is appended data.
```

11. Write a program to copy a file using Files.copy() with REPLACE_EXISTING option.

```
package Assessement_day10;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
public class replace_existing {

public static void main(String[] args) {
Path source = Paths.get("hello.txt");
Path destination = Paths.get("destination.txt");

try {
Files.copy(source, destination,
StandardCopyOption.REPLACE_EXISTING);
```

```
System.out.println("File copied successfully.");
} catch (IOException e) {
System.out.println("Error copying file: " + e.getMessage());
}

}

}
```

Output:

File copied successfully.